NAG Library Routine Document

G05TDF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

G05TDF generates a vector of pseudorandom integers from a discrete distribution with a given PDF (probability density function) or CDF (cumulative distribution function) p.

2 Specification

```
SUBROUTINE G05TDF (MODE, N, P, NP, IP1, ITYPE, R, LR, STATE, X, IFAIL)
INTEGER            MODE, N, NP, IP1, ITYPE, LR, STATE(*), X(N), IFAIL
REAL (KIND=nag_wp) P(NP), R(LR)
```

3 Description

G05TDF generates a sequence of n integers x_i, from a discrete distribution defined by information supplied in P. This may either be the PDF or CDF of the distribution. A reference vector is first set up to contain the CDF of the distribution in its higher elements, followed by an index.

Setting up the reference vector and subsequent generation of variates can each be performed by separate calls to G05TDF or may be combined in a single call.

One of the initialization routines G05KFF (for a repeatable sequence if computed sequentially) or G05KGF (for a non-repeatable sequence) must be called prior to the first call to G05TDF.

4 References

Kendall M G and Stuart A (1969) *The Advanced Theory of Statistics (Volume 1)* (3rd Edition) Griffin

Knuth D E (1981) *The Art of Computer Programming (Volume 2)* (2nd Edition) Addison–Wesley

5 Parameters

1: MODE – INTEGER *Input*

On entry: a code for selecting the operation to be performed by the routine.

MODE = 0
: Set up reference vector only.

MODE = 1
: Generate variates using reference vector set up in a prior call to G05TDF.

MODE = 2
: Set up reference vector and generate variates.

MODE = 3
: Generate variates without using the reference vector.

Constraint: MODE = 0, 1, 2 or 3.

2: N – INTEGER *Input*

On entry: n, the number of pseudorandom numbers to be generated.

Constraint: N $\geq$ 0.

3: P(NP) – REAL (KIND=nag_wp) array *Input*

On entry: the PDF or CDF of the distribution.

Constraints:

$0.0 \leq \mathrm{P}(i) \leq 1.0$, for $i = 1, 2, \ldots, \mathrm{NP}$;
if ITYPE $= 1$, $\sum_{i=1}^{\mathrm{NP}} \mathrm{P}(i) = 1.0$;
if ITYPE $= 2$, $\mathrm{P}(i) < \mathrm{P}(j)$, $i < j$ and $\mathrm{P}(\mathrm{NP}) = 1.0$.

4: NP – INTEGER *Input*

On entry: the number of values supplied in P defining the PDF or CDF of the discrete distribution.

Constraint: NP > 0.

5: IP1 – INTEGER *Input*

On entry: the value of the variate, a whole number, to which the probability in $\mathrm{P}(1)$ corresponds.

6: ITYPE – INTEGER *Input*

On entry: indicates the type of information contained in P.

ITYPE $= 1$
P contains a probability distribution function (PDF).

ITYPE $= 2$
P contains a cumulative distribution function (CDF).

Constraint: ITYPE $= 1$ or 2.

7: R(LR) – REAL (KIND=nag_wp) array *Communication Array*

On entry: if MODE $= 1$, the reference vector from the previous call to G05TDF.

On exit: the reference vector.

8: LR – INTEGER *Input*

On entry: the dimension of the array R as declared in the (sub)program from which G05TDF is called.

Suggested value:

if MODE $\neq 3$, LR $= 10 + 1.4 \times$ NP approximately (for optimum efficiency in generating variates);
otherwise LR $= 1$.

Constraints:

if MODE $= 0$ or 2, LR $\geq$ NP $+ 8$;
if MODE $= 1$, LR should remain unchanged from the previous call to G05TDF.

9: STATE($*$) – INTEGER array *Communication Array*

Note: the actual argument supplied **must** be the array STATE supplied to the initialization routines G05KFF or G05KGF.

On entry: contains information on the selected base generator and its current state.

On exit: contains updated information on the state of the generator.

10: X(N) – INTEGER array *Output*

On exit: contains n pseudorandom numbers from the specified discrete distribution.

11: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value −1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value −1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL = 0 unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL = 0 or −1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL = 1

On entry, MODE = ⟨*value*⟩.
Constraint: MODE = 0, 1 or 2.

IFAIL = 2

On entry, N = ⟨*value*⟩.
Constraint: N ≥ 0.

IFAIL = 3

On entry, at least one element of the vector P is less than 0.0 or greater than 1.0.

On entry, ITYPE = 1 and the sum of the elements of P do not equal one.

On entry, ITYPE = 2 and the values of P are not all in stricly ascending order.

On entry, P(NP) = ⟨*value*⟩.
Constraint: if ITYPE = 2, P(NP) = 1.0.

IFAIL = 4

On entry, NP = ⟨*value*⟩.
Constraint: NP > 0.

IFAIL = 6

On entry, ITYPE = ⟨*value*⟩.
Constraint: ITYPE = 1 or 2.

IFAIL = 7

On entry, some of the elements of the array R have been corrupted or have not been initialized.

The value of NP or IP1 is not the same as when R was set up in a previous call.
Previous value of NP = ⟨*value*⟩ and NP = ⟨*value*⟩.
Previous value of IP1 = ⟨*value*⟩ and IP1 = ⟨*value*⟩.

IFAIL = 8

On entry, LR is too small when MODE = 0 or 2: LR = ⟨*value*⟩, minimum length required = ⟨*value*⟩.

IFAIL $= 9$

On entry, STATE vector has been corrupted or not initialized.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

Not applicable.

8 Parallelism and Performance

G05TDF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

None.

10 Example

This example prints 20 pseudorandom variates from a discrete distribution whose PDF, p, is defined as follows:

n	p
-5	0.01
-4	0.02
-3	0.04
-2	0.08
-1	0.20
0	0.30
1	0.20
2	0.08
3	0.04
4	0.02
5	0.01

The reference vector is set up and and the variates are generated by a single call to G05TDF, after initialization by G05KFF.

10.1 Program Text

```
    Program g05tdfe

!     G05TDF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: g05kff, g05tdf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter              :: lseed = 1, nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                         :: genid, ifail, ip1, itype, lr,        &
                                         lstate, mode, n, np, subid
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable :: p(:), r(:)
      Integer                         :: seed(lseed)
      Integer, Allocatable            :: state(:), x(:)
!     .. Intrinsic Procedures ..
      Intrinsic                       :: ceiling, real
!     .. Executable Statements ..
      Write (nout,*) 'G05TDF Example Program Results'
      Write (nout,*)

!     Skip heading in data file
      Read (nin,*)

!     Read in the base generator information and seed
      Read (nin,*) genid, subid, seed(1)

!     Initial call to initialiser to get size of STATE array
      lstate = 0
      Allocate (state(lstate))
      ifail = 0
      Call g05kff(genid,subid,seed,lseed,state,lstate,ifail)

!     Reallocate STATE
      Deallocate (state)
      Allocate (state(lstate))

!     Initialize the generator to a repeatable sequence
      ifail = 0
      Call g05kff(genid,subid,seed,lseed,state,lstate,ifail)

!     Read in sample size
      Read (nin,*) n

!     Read in the distribution parameters
      Read (nin,*) np, ip1, itype

!     Use suggested value for LR
      lr = 10 + ceiling(1.4E0_nag_wp*real(np,kind=nag_wp))

!     Generate and set up reference vector in one go
      mode = 2

      Allocate (p(np),x(n),r(lr))

!     Read in probabilities
      Read (nin,*) p(1:np)

!     Generate the variates
      ifail = 0
      Call g05tdf(mode,n,p,np,ip1,itype,r,lr,state,x,ifail)
```

```
!     Display the variates
      Write (nout,99999) x(1:n)

99999 Format (1X,I12)
    End Program g05tdfe
```

10.2 Program Data

```
G05TDF Example Program Data
1  1  1762543               :: GENID,SUBID,SEED(1)
20                          :: N
11  -5  1                   :: NP,IP1,ITYPE
0.01 0.02 0.04 0.08 0.2  0.3
0.2  0.08 0.04 0.02 0.01    :: End of P
```

10.3 Program Results

```
 G05TDF Example Program Results

           0
          -2
           1
           1
          -2
           0
           0
           1
           0
           1
          -3
          -1
           0
          -3
           0
          -1
          -1
           5
           2
           0
```